Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №1 по курсу

«Операционные системы»

Группа: М8О-209БВ-24

Студент: Андреев М.В

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 05.10.25

Москва, 2025

Постановка задачи

Вариант 19.

Родительский процесс создает два дочерних процесса. Первой строкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия File с таким именем на запись для child1. Аналогично для второй строки и процесса child2. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс принимает от пользователя строки произвольной длины и пересылает их в pipe1 или в pipe2 в зависимости от правила фильтрации. Процесс child1 и child2 производят работу над строками. Процессы пишут результаты своей работы в стандартный вывод. Правило фильтрации: с вероятностью 80% строки отправляются в pipe1, иначе в pipe2. Дочерние процессы удаляют все гласные из строк.

Общий метод и алгоритм решения

Использованные системные вызовы:

- pid_t fork() - создание дочернего процесса
- int execve(const char *filename, char *const argv[], char *const envp[]) - замена образа памяти процесса
- pid_t waitpid(pid_t pid, int *status, int options) - Ожидание завершения дочернего процесса
- void exit(int status) - завершения выполнения процесса и возвращение статуса
- int pipe(int pipefd[2]) - создание неименованного канала для передачи данных между процессами
- int dup2(int oldfd, int newfd) - переназначение файлового дескриптора
- int open(const char *pathname, int flags, mode_t mode) - открытие\создание файла
- int close(int fd) - закрыть файл

Алгоритм работы программы:

1. **Инициализация**
    - Создание двух каналов (pipe1, pipe2) для межпроцессного взаимодействия
    - Инициализация генератора случайных чисел для правила фильтрации 80/20
2. **Создание процессов**
    - Родительский процесс создает двух дочерних процессов через fork()
    - Каждый дочерний процесс запускает программу ./child_run через execlp()
3. **Распределение данных**

- Пользователь вводит строки в родительский процесс
- Применяется правило фильтрации: 80% строк отправляются в pipe1, 20% - в pipe2

4. **Обработка данных**
   - Дочерние процессы читают строки из своих каналов
   - Удаляют все гласные буквы из полученных строк
   - Записывают результаты в соответствующие файлы
5. **Завершение работы**
   - При вводе "QUIT" родитель закрывает каналы
   - Ожидает завершения дочерних процессов через waitpid()
   - Программа корректно завершает работу

Код программы

parent.c

```
#include "stdio.h"

#include "os_utils.h"
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <time.h>

#define MAX_LINE_LENGTH 1024
#define MAX_FILENAME_LENGTH 256

void start_child_process(int read_fd, const char* child_program, const char*
output_name) {
    if (dup2(read_fd, STDIN_FILENO) == -1) {
        perror("Ошибка dup2");
        exit(IO_ERROR);
    }
    close(read_fd);

    execlp(child_program, child_program, (char* )output_name, (char* )NULL);

    perror("Ошибка execlp");
    exit(EXEC_ERROR);
}

int main() {
```

```
int pipe1_fd[2];
int pipe2_fd[2];

pid_t pid_1, pid_2;

char file1_name[MAX_FILENAME_LENGTH];
char file2_name[MAX_FILENAME_LENGTH];
char temp_buffer[MAX_LINE_LENGTH];

StatusCode status = STATUS_OK;

srand(time(NULL));

const char* str1 = "Введите имя файла для child 1: ";
write(STDOUT_FILENO, "Введите имя файла для child 1: ", strlen(str1));
fflush(stdout);

ssize_t bytes_read = read(STDIN_FILENO, file1_name, MAX_FILENAME_LENGTH - 1);
if (bytes_read <= 0) {
    return INVALID_INPUT;
}
if (file1_name[bytes_read - 1] == '\n') {
    file1_name[bytes_read - 1] = '\0';
} else {
    file1_name[bytes_read] = '\0';
}

const char* str2 = "Введите имя файла для child 2: ";
write(STDOUT_FILENO, "Введите имя файла для child 2: \n", strlen(str2));
fflush(stdout);

bytes_read = read(STDIN_FILENO, file2_name, MAX_FILENAME_LENGTH - 1);
if (bytes_read <= 0) {
    return INVALID_INPUT;
}
if (file2_name[bytes_read - 1] == '\n') {
    file2_name[bytes_read - 1] = '\0';
} else {
    file2_name[bytes_read] = '\0';
}

if (pipe(pipe1_fd) == -1 || pipe(pipe2_fd) == -1) {
    perror("Ошибка pipe");
    return PIPE_ERROR;
}

pid_1 = fork();
if (pid_1 == -1) {
    perror("Ошибка fork 1");
```

```
    status = FORK_ERROR;
    goto cleanup_pipes;
}

if (pid_1 == 0) {
    close(pipe1_fd[1]);
    close(pipe2_fd[0]);
    close(pipe2_fd[1]);

    start_child_process(pipe1_fd[0], "./child_run", file1_name);
}

pid_2 = fork();
if (pid_2 == -1) {
    perror("Ошибка fork 2");
    status = FORK_ERROR;
    goto cleanup_pipes;
}

if (pid_2 == 0) {
    close(pipe2_fd[1]);
    close(pipe1_fd[0]);
    close(pipe1_fd[1]);

    start_child_process(pipe2_fd[0], "./child_run", file2_name);
}

close(pipe1_fd[0]);
close(pipe2_fd[0]);

const char* par_start = "Родительский процесс начался \n";
write(STDOUT_FILENO, "Родительский процесс начался \n", strlen(par_start));
const char* input_str = "Введите строчки \n";
write(STDOUT_FILENO, "Введите строчки \n", strlen(input_str));
fflush(stdout);

char buffer[MAX_LINE_LENGTH];
while (fgets(buffer, sizeof(buffer), stdin) != NULL) {
    int len = strlen(buffer);
    if (len > 0 && buffer[len - 1] == '\n') {
        buffer[len - 1] = '\0';
        len--;
    }
    if (strcmp(buffer, "QUIT") == 0) {
        break;
    }

    int write_fd = -1;
```

```
        int random_percent = rand() % 100;

        if (random_percent < 80) {
            write_fd = pipe1_fd[1];
            int n = snprintf(temp_buffer, sizeof(temp_buffer), "Отправлено в Child 1
(вероятность: %d%%)\n", random_percent);
            write(STDOUT_FILENO, temp_buffer, n);
        } else {
            write_fd = pipe2_fd[1];
            int n = snprintf(temp_buffer, sizeof(temp_buffer), "Отправлено в Child 2
(вероятность: %d%%)\n", random_percent);
            write(STDOUT_FILENO, temp_buffer, n);
        }
        fflush(stdout);

        if (len < sizeof(buffer) - 1) {
            buffer[len] = '\n';
            buffer[len + 1] = '\0';
            len++;
        }

        if (write(write_fd, buffer, len) != len) {
            perror("Ошибка с pipe");
            return PIPE_ERROR;
        }
    }

    close(pipe1_fd[1]);
    close(pipe2_fd[1]);

    waitpid(pid_1, NULL, 0);
    waitpid(pid_2, NULL, 0);

    const char* end_msg = "Родительский и детский процесс успешно завершены\n";
    write(STDOUT_FILENO, "Родительский и детский процесс успешно завершены\n",
strlen(end_msg));
    fflush(stdout);

    return STATUS_OK;

cleanup_pipes:
    close(pipe1_fd[0]); close(pipe1_fd[1]);
    close(pipe2_fd[0]); close(pipe2_fd[1]);
    return status;

}
```

child.c

```c
#include "os_utils.h"
#include <ctype.h>

void remove_vowels(char* str) {
    if (!str) {
        return;
    }
    int write_idx = 0;
    for (int read_idx = 0; str[read_idx] != '\0'; read_idx++) {
        char c = str[read_idx];
        char lower_c = tolower((unsigned char) c);
        if (lower_c != 'a' && lower_c != 'e' && lower_c != 'i' && lower_c != 'o' &&
lower_c != 'u' && lower_c != 'y') {
            str[write_idx++] = c;
        }
    }
    str[write_idx] = '\0';
}

int main(int argc, char* argv[]) {
    if (argc != 2) {
        return INVALID_INPUT;
    }
    const char* output_name = argv[1];
    FILE* output = NULL;
    char buffer[1024];

    output = fopen(output_name, "w");
    if (output == NULL) {
        return FILE_OPEN_ERROR;
    }
    while (fgets(buffer, sizeof(buffer), stdin) != NULL) {
        remove_vowels(buffer);
        if (fputs(buffer, output) == EOF) {
            return IO_ERROR;
        }
    }
    fclose(output);
    return STATUS_OK;
}
```

os_utils.h

```c
#ifndef OS_UTILS_H
#define OS_UTILS_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>
```

```c
#include <errno.h>

typedef enum {
    STATUS_OK = 0,
    PIPE_ERROR = 1,
    FORK_ERROR = 2,
    EXEC_ERROR = 3,
    INVALID_INPUT = 4,
    FILE_OPEN_ERROR = 5,
    IO_ERROR = 6
} StatusCode;

void remove_vowels(char* str);

#endif
```

Протокол работы программы

Тестирование:

```
maxva@LAPTOP-RQH0OLUE:/mnt/d/Programming/OS/lab1$ gcc -o parent parent.c
maxva@LAPTOP-RQH0OLUE:/mnt/d/Programming/OS/lab1$ gcc -o child_run child.c
maxva@LAPTOP-RQH0OLUE:/mnt/d/Programming/OS/lab1$ ls -la parent child_run
-rwxrwxrwx 1 maxva maxva 16264 Oct  7 00:08 child_run
-rwxrwxrwx 1 maxva maxva 16840 Oct  7 00:08 parent
maxva@LAPTOP-RQH0OLUE:/mnt/d/Programming/OS/lab1$ ./parent
Введите имя файла для child 1: file1.txt
Введите имя файла для child 2: file2.txt
Родительский процесс начался
Введите строчки
Hello World
Отправлено в Child 2 (вероятность: 90%)
Test string
Отправлено в Child 1 (вероятность: 16%)
QUIT
Родительский и детский процесс успешно завершены
maxva@LAPTOP-RQH0OLUE:/mnt/d/Programming/OS/lab1$ cat file1.txt
Tst strng
maxva@LAPTOP-RQH0OLUE:/mnt/d/Programming/OS/lab1$ cat file2.txt
Hll Wrld
```

Strace:

```
$ strace -f ./parent
execve("./parent", ["./parent"], 0x7ffd362c8648 /* 36 vars */) = 0
brk(NULL)                               = 0x5609e67bf000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f3638ce1000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
```

```
fstat(3, {st_mode=S_IFREG|0644, st_size=19131, ...}) = 0
mmap(NULL, 19131, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f3638cdc000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0"..., 832)
= 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f3638aca000
mmap(0x7f3638af2000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f3638af2000
mmap(0x7f3638c7a000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x7f3638c7a000
mmap(0x7f3638cc9000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1fe000) = 0x7f3638cc9000
mmap(0x7f3638ccf000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f3638ccf000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f3638ac7000
arch_prctl(ARCH_SET_FS, 0x7f3638ac7740) = 0
set_tid_address(0x7f3638ac7a10)         = 126225
set_robust_list(0x7f3638ac7a20, 24)     = 0
rseq(0x7f3638ac8060, 0x20, 0, 0x53053053) = 0
mprotect(0x7f3638cc9000, 16384, PROT_READ) = 0
mprotect(0x5609cb774000, 4096, PROT_READ) = 0
mprotect(0x7f3638d19000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f3638cdc000, 19131)           = 0
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\270\320\274\321\217 \321\204\320\260\320\271\320\273\320\260"..., 49Введите имя файла
для child 1: ) = 49
read(0, f_1.txt
"f_1.txt\n", 255)               = 8
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\270\320\274\321\217 \321\204\320\260\320\271\320\273\320\260"..., 49Введите имя файла
для child 2: ) = 49
read(0, f_2.txt
"f_2.txt\n", 255)               = 8
pipe2([3, 4], 0)                        = 0
pipe2([5, 6], 0)                        = 0
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace:
Process 126319 attached
, child_tidptr=0x7f3638ac7a10) = 126319
[pid 126319] set_robust_list(0x7f3638ac7a20, 24 <unfinished ...>
[pid 126225] clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD <unfinished ...>
[pid 126319] <... set_robust_list resumed>) = 0
[pid 126319] close(4strace: Process 126320 attached
)                       = 0
[pid 126225] <... clone resumed>, child_tidptr=0x7f3638ac7a10) = 126320
[pid 126320] set_robust_list(0x7f3638ac7a20, 24 <unfinished ...>
[pid 126225] close(3 <unfinished ...>
[pid 126319] close(5 <unfinished ...>
[pid 126225] <... close resumed>)       = 0
```

```
[pid 126320] <... set_robust_list resumed>) = 0
[pid 126225] close(5 <unfinished ...>
[pid 126319] <... close resumed>)       = 0
[pid 126320] close(6 <unfinished ...>
[pid 126319] close(6 <unfinished ...>
[pid 126225] <... close resumed>)       = 0
[pid 126320] <... close resumed>)       = 0
[pid 126225] write(1,
"\320\240\320\276\320\264\320\270\321\202\320\265\320\273\321\214\321\201\320\272\320\270\320
0\271 \320\277\321\200\320\276\321"..., 56 <unfinished ...>
Родительский процесс начался
[pid 126319] <... close resumed>)       = 0
[pid 126225] <... write resumed>)       = 56
[pid 126320] close(3 <unfinished ...>
[pid 126225] write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\321\207\320\272\320\270 \n", 31 <unfinished ...>
[pid 126319] dup2(3, 0Введите строчки
 <unfinished ...>
[pid 126225] <... write resumed>)       = 31
[pid 126320] <... close resumed>)       = 0
[pid 126225] fstat(0,  <unfinished ...>
[pid 126319] <... dup2 resumed>)        = 0
[pid 126225] <... fstat resumed>{st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x6),
...}) = 0
[pid 126320] close(4 <unfinished ...>
[pid 126225] getrandom( <unfinished ...>
[pid 126319] close(3 <unfinished ...>
[pid 126320] <... close resumed>)       = 0
[pid 126225] <... getrandom resumed>"\x31\x95\x3c\x35\x8a\xaf\x0e\xca", 8,
GRND_NONBLOCK) = 8
[pid 126319] <... close resumed>)       = 0
[pid 126320] dup2(5, 0 <unfinished ...>
[pid 126319] execve("./child_run", ["./child_run", "f_1.txt"], 0x7ffe1c3775b8 /* 36
vars */ <unfinished ...>
[pid 126225] brk(NULL <unfinished ...>
[pid 126320] <... dup2 resumed>)        = 0
[pid 126225] <... brk resumed>)         = 0x5609e67bf000
[pid 126320] close(5 <unfinished ...>
[pid 126225] brk(0x5609e67e0000 <unfinished ...>
[pid 126320] <... close resumed>)       = 0
[pid 126225] <... brk resumed>)         = 0x5609e67e0000
[pid 126320] execve("./child_run", ["./child_run", "f_2.txt"], 0x7ffe1c3775b8 /* 36
vars */ <unfinished ...>
[pid 126225] read(0,  <unfinished ...>
[pid 126320] <... execve resumed>)      = 0
[pid 126319] <... execve resumed>)      = 0
[pid 126320] brk(NULL <unfinished ...>
[pid 126319] brk(NULL <unfinished ...>
[pid 126320] <... brk resumed>)         = 0x556b9a4c3000
[pid 126319] <... brk resumed>)         = 0x555ea04cc000
[pid 126320] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x7fac916dc000
[pid 126319] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
```

```
[pid 126320] access("/etc/ld.so.preload", R_OK <unfinished ...>
[pid 126319] <... mmap resumed>)        = 0x7f7eb66fc000
[pid 126320] <... access resumed>)      = -1 ENOENT (No such file or directory)
[pid 126319] access("/etc/ld.so.preload", R_OK <unfinished ...>
[pid 126320] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
[pid 126319] <... access resumed>)      = -1 ENOENT (No such file or directory)
[pid 126320] fstat(3,  <unfinished ...>
[pid 126319] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>
[pid 126320] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=19131, ...}) = 0
[pid 126319] <... openat resumed>)      = 3
[pid 126320] mmap(NULL, 19131, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>
[pid 126319] fstat(3,  <unfinished ...>
[pid 126320] <... mmap resumed>)        = 0x7fac916d7000
[pid 126319] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=19131, ...}) = 0
[pid 126320] close(3 <unfinished ...>
[pid 126319] mmap(NULL, 19131, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>
[pid 126320] <... close resumed>)       = 0
[pid 126319] <... mmap resumed>)        = 0x7f7eb66f7000
[pid 126320] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC
<unfinished ...>
[pid 126319] close(3 <unfinished ...>
[pid 126320] <... openat resumed>)      = 3
[pid 126319] <... close resumed>)       = 0
[pid 126320] read(3,  <unfinished ...>
[pid 126319] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC
<unfinished ...>
[pid 126320] <... read
resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0"..., 832) = 832
[pid 126319] <... openat resumed>)      = 3
[pid 126320] pread64(3,  <unfinished ...>
[pid 126319] read(3,  <unfinished ...>
[pid 126320] <... pread64
resumed>"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
[pid 126319] <... read
resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0"..., 832) = 832
[pid 126320] fstat(3,  <unfinished ...>
[pid 126319] pread64(3,  <unfinished ...>
[pid 126320] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0
[pid 126319] <... pread64
resumed>"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
[pid 126320] pread64(3,  <unfinished ...>
[pid 126319] fstat(3,  <unfinished ...>
[pid 126320] <... pread64
resumed>"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
[pid 126319] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0
[pid 126320] mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fac914c5000
[pid 126319] pread64(3,  <unfinished ...>
[pid 126320] mmap(0x7fac914ed000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000 <unfinished ...>
[pid 126319] <... pread64
resumed>"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
[pid 126320] <... mmap resumed>)        = 0x7fac914ed000
[pid 126320] mmap(0x7fac91675000, 323584, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000 <unfinished ...>
```

```
[pid 126319] mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0 <unfinished ...>
[pid 126320] <... mmap resumed>)        = 0x7fac91675000
[pid 126319] <... mmap resumed>)        = 0x7f7eb64e5000
[pid 126320] mmap(0x7fac916c4000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000 <unfinished ...>
[pid 126319] mmap(0x7f7eb650d000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000 <unfinished ...>
[pid 126320] <... mmap resumed>)        = 0x7fac916c4000
[pid 126319] <... mmap resumed>)        = 0x7f7eb650d000
[pid 126320] mmap(0x7fac916ca000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 126319] mmap(0x7f7eb6695000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000 <unfinished ...>
[pid 126320] <... mmap resumed>)        = 0x7fac916ca000
[pid 126319] <... mmap resumed>)        = 0x7f7eb6695000
[pid 126320] close(3)                   = 0
[pid 126319] mmap(0x7f7eb66e4000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000 <unfinished ...>
[pid 126320] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 126319] <... mmap resumed>)        = 0x7f7eb66e4000
[pid 126320] <... mmap resumed>)        = 0x7fac914c2000
[pid 126319] mmap(0x7f7eb66ea000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 126320] arch_prctl(ARCH_SET_FS, 0x7fac914c2740 <unfinished ...>
[pid 126319] <... mmap resumed>)        = 0x7f7eb66ea000
[pid 126320] <... arch_prctl resumed>)  = 0
[pid 126319] close(3 <unfinished ...>
[pid 126320] set_tid_address(0x7fac914c2a10 <unfinished ...>
[pid 126319] <... close resumed>)       = 0
[pid 126320] <... set_tid_address resumed>) = 126320
[pid 126319] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 126320] set_robust_list(0x7fac914c2a20, 24 <unfinished ...>
[pid 126319] <... mmap resumed>)        = 0x7f7eb64e2000
[pid 126320] <... set_robust_list resumed>) = 0
[pid 126319] arch_prctl(ARCH_SET_FS, 0x7f7eb64e2740 <unfinished ...>
[pid 126320] rseq(0x7fac914c3060, 0x20, 0, 0x53053053 <unfinished ...>
[pid 126319] <... arch_prctl resumed>)  = 0
[pid 126320] <... rseq resumed>)        = 0
[pid 126319] set_tid_address(0x7f7eb64e2a10 <unfinished ...>
[pid 126320] mprotect(0x7fac916c4000, 16384, PROT_READ <unfinished ...>
[pid 126319] <... set_tid_address resumed>) = 126319
[pid 126320] <... mprotect resumed>)    = 0
[pid 126319] set_robust_list(0x7f7eb64e2a20, 24 <unfinished ...>
[pid 126320] mprotect(0x556b979ac000, 4096, PROT_READ <unfinished ...>
[pid 126319] <... set_robust_list resumed>) = 0
[pid 126320] <... mprotect resumed>)    = 0
[pid 126319] rseq(0x7f7eb64e3060, 0x20, 0, 0x53053053 <unfinished ...>
[pid 126320] mprotect(0x7fac91714000, 8192, PROT_READ <unfinished ...>
[pid 126319] <... rseq resumed>)        = 0
[pid 126320] <... mprotect resumed>)    = 0
[pid 126320] prlimit64(0, RLIMIT_STACK, NULL,  <unfinished ...>
```

```
[pid 126319] mprotect(0x7f7eb66e4000, 16384, PROT_READ <unfinished ...>
[pid 126320] <... prlimit64 resumed>{rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) =
0
[pid 126319] <... mprotect resumed>)    = 0
[pid 126320] munmap(0x7fac916d7000, 19131 <unfinished ...>
[pid 126319] mprotect(0x555e9f442000, 4096, PROT_READ <unfinished ...>
[pid 126320] <... munmap resumed>)      = 0
[pid 126319] <... mprotect resumed>)    = 0
[pid 126320] getrandom("\x90\x20\x2e\xd2\x1f\xaf\xae\x3f", 8, GRND_NONBLOCK) = 8
[pid 126320] brk(NULL)                  = 0x556b9a4c3000
[pid 126320] brk(0x556b9a4e4000)        = 0x556b9a4e4000
[pid 126320] openat(AT_FDCWD, "f_2.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666 <unfinished
...>
[pid 126319] mprotect(0x7f7eb6734000, 8192, PROT_READ) = 0
[pid 126319] prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
[pid 126319] munmap(0x7f7eb66f7000, 19131) = 0
[pid 126319] getrandom("\xab\xf2\x19\x9a\xe4\xe7\x25\xd7", 8, GRND_NONBLOCK) = 8
[pid 126319] brk(NULL)                  = 0x555ea04cc000
[pid 126319] brk(0x555ea04ed000)        = 0x555ea04ed000
[pid 126319] openat(AT_FDCWD, "f_1.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666 <unfinished
...>
[pid 126320] <... openat resumed>)      = 3
[pid 126320] fstat(0, {st_mode=S_IFIFO|0600, st_size=0, ...}) = 0
[pid 126320] read(0,  <unfinished ...>
[pid 126319] <... openat resumed>)      = 3
[pid 126319] fstat(0, {st_mode=S_IFIFO|0600, st_size=0, ...}) = 0
[pid 126319] read(0, Hello World
 <unfinished ...>
[pid 126225] <... read resumed>"Hello World\n", 1024) = 12
[pid 126225] write(1,
"\320\236\321\202\320\277\321\200\320\260\320\262\320\273\320\265\320\275\320\276 \320\262
Child 1 ..., 62Отправлено в Child 1 (вероятность: 62%)
) = 62
[pid 126225] write(4, "Hello World\n", 12) = 12
[pid 126319] <... read resumed>"Hello World\n", 4096) = 12
[pid 126225] read(0,  <unfinished ...>
[pid 126319] fstat(3, {st_mode=S_IFREG|0777, st_size=0, ...}) = 0
[pid 126319] read(0, Hi
 <unfinished ...>
[pid 126225] <... read resumed>"Hi\n", 1024) = 3
[pid 126225] write(1,
"\320\236\321\202\320\277\321\200\320\260\320\262\320\273\320\265\320\275\320\276 \320\262
Child 2 ..., 62Отправлено в Child 2 (вероятность: 97%)
) = 62
[pid 126225] write(6, "Hi\n", 3)        = 3
[pid 126320] <... read resumed>"Hi\n", 4096) = 3
[pid 126225] read(0,  <unfinished ...>
[pid 126320] fstat(3, {st_mode=S_IFREG|0777, st_size=0, ...}) = 0
[pid 126320] read(0, QUIT
 <unfinished ...>
[pid 126225] <... read resumed>"QUIT\n", 1024) = 5
[pid 126225] close(4)                   = 0
[pid 126319] <... read resumed>"", 4096) = 0
```

```
[pid 126225] close(6)                   = 0
[pid 126320] <... read resumed>"", 4096) = 0
[pid 126319] write(3, "Hll Wrld\n", 9 <unfinished ...>
[pid 126225] wait4(126319,  <unfinished ...>
[pid 126320] write(3, "H\n", 2 <unfinished ...>
[pid 126319] <... write resumed>)       = 9
[pid 126319] close(3 <unfinished ...>
[pid 126320] <... write resumed>)       = 2
[pid 126320] close(3)                   = 0
[pid 126319] <... close resumed>)       = 0
[pid 126320] exit_group(0 <unfinished ...>
[pid 126319] exit_group(0 <unfinished ...>
[pid 126320] <... exit_group resumed>)  = ?
[pid 126319] <... exit_group resumed>)  = ?
[pid 126320] +++ exited with 0 +++
[pid 126225] <... wait4 resumed>NULL, 0, NULL) = ? ERESTARTSYS (To be restarted if
SA_RESTART is set)
[pid 126225] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=126320,
si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---
[pid 126225] wait4(126319,  <unfinished ...>
[pid 126319] +++ exited with 0 +++
<... wait4 resumed>NULL, 0, NULL)       = 126319
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=126319, si_uid=1000,
si_status=0, si_utime=0, si_stime=0} ---
wait4(126320, NULL, 0, NULL)            = 126320
write(1,
"\320\240\320\276\320\264\320\270\321\202\320\265\320\273\321\214\321\201\320\272\320\270\32
0\271 \320\270 \320\264\320\265"..., 92Родительский и детский процесс успешно завершены
) = 92
exit_group(0)                           = ?
+++ exited with 0 +++
```

Вывод

В ходе выполнения лабораторной работы были успешно освоены принципы управления процессами и организации межпроцессного взаимодействия в операционных системах. Основные трудности возникли при отладке программы, особенно с корректным закрытием файловых дескрипторов каналов и обеспечением синхронизации между родительским и дочерними процессами. В процессе работы научился использовать системные вызовы fork(), pipe(), execve() и другие, что позволило глубже понять механизмы работы операционной системы с процессами и каналами. Полученные навыки будут полезны для выполнения последующих лабораторных работ по данной дисциплине.